IN THE CLAIMS

1 (Original). A method comprising:

assigning a number of stall cycles between a first and a second instruction; and

scheduling said first and second instructions for execution based on the assigned stall cycles.

2 (Currently Amended). The method of claim 1, further comprising:

using a ~~the~~ number of maximum possible pipeline stall cycles between said first and second instructions to indicate a ~~the~~ data dependency therebetween.

3 (Original). The method of claim 2, further comprising:

extending a register scoreboard that keeps track of the data dependency.

4 (Original). The method of claim 3, further comprising:

maintaining a count of issue latency for said first and second instructions.

5 (Currently Amended). The method of claim 3, further comprising:

maintaining a count for a ~~the~~ number of cycles from start to end of a ~~the~~ issue of said first and second instructions.

6 (Original). The method of claim 3, further comprising:

maintaining a count for pipeline stalls between said first instruction and a previous instruction.

7 (Currently Amended). The method of claim 3, further comprising:

extending the register scoreboard by m rows and m columns to keep track of a ~~the~~ maximum possible pipeline stall cycles.

8 (Currently Amended). The method of claim 7, further comprising:

keeping track of a ~~the~~ first non-zero value from right to left in an ~~the~~ m-th row of the register scoreboard to reorder said first instruction.

9 (Currently Amended). The method of claim 7, further comprising:

keeping track of a ~~the~~ first non-zero value from top to bottom in an ~~the~~ m-th column of the register scoreboard to reorder said first instruction.

10 (Original). The method of claim 3, further comprising:

keeping track of an instruction that causes pipeline stall.

11 (Currently Amended). An apparatus comprising:

a register to store a number of stall cycles between a first and a second instruction; and

a compiler coupled to schedule said first and second instructions for execution based on the ~~assigned~~ stall cycles.

12 (Currently Amended). The apparatus of claim 11, wherein said compiler uses a ~~the~~ number of maximum possible pipeline stall cycles between said first and second instructions to indicate data dependency therebetween.

13 (Original). The apparatus of claim 12, wherein said register is extended by m-rows and m-columns to keep track of maximum possible pipeline stall cycles.

14 (Currently Amended). The apparatus of claim 13, wherein said compiler to keep track of a ~~the~~ first non-zero value from right to left in m-th row to reorder said first instruction.

15 (Currently Amended). The apparatus of claim 13, wherein said compiler to keep track of a ~~the~~ first non-zero value from top to bottom in the m-th column to reorder the first instruction.

16 (Original). A system comprising:

a non-volatile storage storing instructions;

a processor to execute at least some of the instructions to provide a virtual machine that assigns a number of stall cycles between a first and a second instruction and schedules said first and second instructions for execution based on the assigned stall cycles.

17 (Original). The system of claim 16, further comprising:

a register to store dependency data between said first and second instructions.

18 (Currently Amended). The system of claim 17, further comprising:

a compiler coupled to schedule said first and second instructions for execution based on a ~~the~~ maximum possible pipeline stall cycles.

19 (Original). The system of claim 16, wherein said register is a register scoreboard.

20 (Original). The system of claim 17, wherein said compiler is just-in-time compiler for an object-oriented programming language.

21 (Original). An article comprising a computer readable storage medium storing instructions that, when executed cause a processor-based system to:

assign a number of stall cycles between a first and a second instruction; and

schedule said first and second instructions for execution based on the assigned stall cycles.

22 (Original). The article of claim 21, comprising a medium storing instructions that, when executed cause a processor-based system to:

use the number of maximum possible pipeline stall cycles between said first and second instructions to indicate the data dependency therebetween.

23 (Original). The article of claim 22, comprising a medium storing instructions that, when executed cause a processor-based system to:

extend a register scoreboard that keeps track of the data dependency.

24 (Original). The article of claim 23, comprising a medium storing instructions that, when executed cause a processor-based system to:

maintain a count of issue latency for said first and second instructions.

25 (Original). The article of claim 23, comprising a medium storing instructions that, when executed cause a processor-based system to:

maintain a count for the number of cycles from start to end of the issue of said first and second instructions.

26 (Original). The article of claim 23, comprising a medium storing instructions that, when executed cause a processor-based system to:

maintain a count for pipeline stalls between said first instruction and a previous instruction.

27 (Original). The article of claim 23, comprising a medium storing instructions that, when executed cause a processor-based system to:

extend the register scoreboard by m rows and m columns to keep track of the maximum possible pipeline stall cycles.

28 (Original). The article of claim 27, comprising a medium storing instructions that, when executed cause a processor-based system to:

keep track of the first non-zero value from right to left in the m-th row of the register scoreboard to reorder said first instruction.

29 (Original). The article of claim 27, comprising a medium storing instructions that, when executed cause a processor-based system to:

keep track of the first non-zero value from top to bottom in the m-th column of the register scoreboard to reorder said first instruction.

30 (Original). The article of claim 23, comprising a medium storing instructions that, when executed cause a processor-based system to:

keep track of an instruction that causes pipeline stall.